WEST

End of Result Set

Generate Collection

L7: Entry 1 of 1 File: USPT May 9, 1995

DOCUMENT-IDENTIFIER: US 5414833 A
TITLE: Network security system and method using a parallel finite state machine adaptive active monitor and responder

BSPR:
This patent application is also related to the copending U.S. patent application, Ser. No. 08/138,045, filed Oct. 15, 1993, entitled "System and Method for Adaptive, Active Monitoring of a Serial Data Stream Having a Characteristic Pattern," by P. C. Hershey, et al., assigned to the IBM Corporation and incorporated herein by reference.

BSPR:
The co-pending patent application by Hershey et al. entitled "System and Method for Adaptive, Active Monitoring of a Serial Data Stream Having a Characteristic Pattern", Ser. No. 08/138,045 describes a programmable method for detecting characteristic data patterns of diverse size transmitted over high-speed data links. Unlike more traditional method in data is sampled and stored in a log, the finite state machine (FSM) information monitoring means of the Hershey invention, cited above, is programmed to "look" for interesting patterns of concern. In this way, the FSM discards most of the high-speed information bits and concentrates only on patterns of interest. In short, the FSM signals a pattern match as opposed to collecting and storing data in a log, which must then be processed by some other network function. The FSM information monitoring means is coupled to the network, and in response to detecting a prescribed pattern, outputs a control signal to the network to alter communication characteristics thereof. How this control signal is handled depends on the application supported by the FSM information monitoring means. The Hershey et al. application, Ser. No. 08/138,045 is limited in its teaching of how systems can respond to the detection of prescribed patterns, concentrating more of the problem of pattern detection itself. Moreover, the Hershey application, referred to above as well as other prior art, does not teach a unified method for (a) monitoring of security events--virus patterns, natural language patterns, and intrusion detection patterns--on high-speed communication links, (b) reporting detected security events to a network security manager, and (c) responding on a network-level to detected security events as a means to thwart, counter, minimize, or isolate their possible harmful effects.

BSPR:
Viruses can be detected by two primary means: (1) modification detection and (2) pattern detection via a scanner. With modification detection, checksums or cryptographic hash values are used to detect changes in executable codes. These changes are reported to a system manager who then decides whether the change is expected (e.g., due to a recent software upgrade) or unexpected (e.g., due to viral infection or unauthorized modification). This method usually requires manual intervention to add, delete, or modify system files in order to ensure adequate coverage and to limit the number of false alarms. A list of checksums must be maintained for all files to be protected. This method is not practical in a high-speed communications environment for several reasons: (1) the overhead imposed by computing checksums, (2) the unpredictability of data flowing on the communications medium, and (3) the requirement for transporting and storing reference checksums for use in comparing with the computed checksums.

BSPR:
With pattern detection via a scanner, system files are periodically scanned for patterns, which consist of a set of pre-defined virus "signatures." Pattern matches are reported to the system manager who then decides whether the match represents a misdiagnosis or an actual viral infection. A virus consists of one or more fixed-length signature patterns, so the number of virus signatures is proportional to the number of viruses. A list of virus signatures must be maintained for each virus. The pattern search usually proceeds in a serial fashion, scanning each file one at a time, comparing the records of the file with each signature pattern in turn. This form of pattern detection is not suitable for a high speed communications environment because of the delay caused by the serial, fixed-signature search pattern. In a high speed communications environment, it would be desirable to search for many different signature patterns in parallel.

BSPR:
An article entitled "Towards a Testbed for Malicious Code Detection," by R. Lo, P. Kerchen, R. Crawford, W. Ho, and J. Crossley, Lawrence Livermore National Lab Report UCRL-JC-105792, 1991, which describes static and dynamic analysis tools which have been shown to be effective against certain types of malicious code. Such an idea represents another form of anomaly detection.

BSPR:
The described methods of virus detection (modification detection and pattern detection) are based on identifying a viral infection in a stored form of the data--after the infection has already taken place. A different, highly parallel method is required in order to detect the transfer of viral agents across a high-speed communications link where one "looks" for a viral pattern as it flashes past a monitor attached to the bit stream. The Hershey, et al. adaptive, active monitor described in copending U.S. patent application, Ser. No. 08/138,045, filed Oct. 15, 1993, entitled "System and

Method for Adaptive, Active Monitoring of a Serial Data Stream Having a Characteristic Pattern," cited above in Related Patents and Patent Applications, is particularly well-suited for scanning of virus signatures in a high-speed communications environment. The prior art with respect to virus detection, virus scanning, signature preparation, virus reporting, etc. is well defined and described as pointed out above. However, the prior art does not teach how to construct a virus scanning apparatus well-suited to very high-speed networks, and more particularly how such an apparatus could be constructed for attachment to a bit stream as a singular entity, integrated within a network-attached device or standing alone, whose purpose is to act as a monitoring and responding device for assuring the integrity and security of a high-speed communications network.

BSPR:
Each of the natural language detection applications has a range of possible actions that may be taken in response to a detected offending pattern.

BSPR:
One form of intrusion of particular concern to network security is an adversary who attempts to gain access to a system by issuing repeated login requests. In this case, intrusion detection is aimed at detecting a higher-than-normal frequency of login sequences indicating that someone is repeatedly attempting to login by guessing userids and passwords. In this security application, one does not merely detect the presence of a pattern but the presence of a higher-than-normal frequency of patterns.

BSPR:
Most security applications (including virus detection, natural language detection, and intrusion detection) consist of a detection step and a response step. The Hershey FSM information monitoring means described in U.S. pending patent application Ser. No. 08/138,045, cited above, is particularly suited as a pattern detection means in a high-speed communication environment. Yet for the Hershey FSM information monitoring means to be well-suited as a security device in a high-speed communication environment, it must be adapted to search for patterns particular to security applications and it must be extended to provide a capability for responding, in appropriate ways, to detected patterns. Such a security device (hereinafter called a security agent) must provide both an information monitoring function as well as a real-time responding function.

DRPR:
FIG. 1D illustrate a first finite state machine serving as the first patterns analysis stage in a plurality of finite state machines.

DRPR:
FIG. 6 is a block diagram illustration of the Responder 300 that processes security events, which are characterized as one

of a plurality of possible pattern alarms (144, 144', ..., 144") output by Adaptive, Active Monitor 100 of FIG. 4.

DRPR:
FIG. 8 is a block diagram illustration of an alternate embodiment of the invention (as described in FIG. 6) wherein the pattern alarms 144a, 144b, ..., 144n from the Hershey adaptive, active monitor 100 are applied to counters 360a, 360b, ..., 360n, respectively, in order to prevent adaptive, active monitor 100 (see FIG. 4) of FIG. 4 from over-running responder 300.

DRPR:
FIG. 9 depicts an extended security alert message 341 consisting of a security agent identifier 321, a security code 322, a sequence number counter 325, and a pattern alarm counter value 326.

DRPR:
FIG. 10 is a block diagram illustration of another alternate embodiment of the invention wherein the Hershey adaptive, active monitor 100 of FIG. 4 is configured as an intrusion detector and responder 300 is designed to produce and transmit a security alert message whenever the number of detected pattern alarms of a particular type in a given interval of time reaches a prescribed threshold value.

DRPR:
FIG. 12 is an example embodiment of pattern alarms and counters of FIG. 8 configured for virus detection.

DEPR:
In accordance with the invention, a security agent incorporates both (1) an adaptive, active monitoring means and (2) a responding means. The adaptive, active monitoring means is based on the adaptive, active monitoring means taught in copending patent application by Paul Hershey and John Waclawsky, entitled 'System and Method for Adaptive, Active Monitoring of a Serial Data Stream Having a Characteristic Pattern', Ser. No. 08/138,045 cited above under Related Patents and Patent Applications, assigned to the IBM Corporation and incorporated herein by reference. The responding means, when coupled to the adaptive, active monitoring means, provides the functionality necessary to implement a security agent in a high-speed communication environment.

DEPR:
The adaptive, active monitor is useful in detecting characteristic data patterns in messages on a high-speed data network, such as starting delimiters, tokens, various types of frames, and protocol information. Such serial data streams include serial patterns of binary bits, and can also include serial patterns of multiple state symbols, such as the J, K, 0, 1 symbols (four states) of token ring networks. Such serial data streams can further include multiple state symbols in fiber optical distributed data interface (FDDI) networks.

DEPR:
Characteristic data patterns such as these, include component bit patterns, some of which are common among several characteristic data patterns. For example, a starting delimiter bit pattern is a common component which begins many other characteristic data patterns such as a token, a MAC frame, and a beacon frame in the IEEE 802.5 token ring protocol. The occurrence of multiple component bit patterns in a characteristic data pattern can be generalized by referring to a first component pattern which is followed by a second component pattern.

DEPR:
The adaptive, active monitor comprises two finite state machines (FSM) which are constructed to detect the occurrence of a characteristic data pattern having two consecutive component bit patterns. The first FSM is called the predecessor FSM, and it is configured to detect the first component pattern. The second FSM is called the successor FSM, and it is configured to detect the second component pattern. The first FSM will send a starting signal to the second FSM, when the first FSM has successfully detected the first component pattern. The starting signal initializes the second FSM, to take over the analysis of the portion of the bit stream which follows the first component pattern. If the second FSM successfully detects the second component pattern, it then outputs a pattern alarm signal, indicating the successful detection of the entire characteristic data pattern.

DEPR:
Another feature of the adaptive, active monitor is the accommodation of a component bit pattern which is common to two or more distinctly different characteristic data patterns. For example, a first characteristic data pattern is composed of a first-type component bit pattern followed by a second-type component bit pattern. A second characteristic data pattern is composed of the same first-type component bit pattern followed by a third type component bit pattern. A first FSM is configured to detect the first component pattern, a second FSM is configured to detect the second component pattern, and a third FSM is configured to detect the third component pattern. The objective is to detect either one of the two characteristic data patterns. The predecessor FSM will have a plurality of successor FSMs which run simultaneously in parallel. The first FSM will send a starting signal to both the second FSM and to the third FSM, when the first FSM has successfully detected the first component pattern. The starting signal initializes the second FSM, to take over the analysis of the bit stream which follows the first component pattern, to look for the second component bit pattern. And the starting signal initializes the third FSM, to take over the analysis of the same bit stream which follows the first component pattern, to look for the third component bit pattern. The second FSM and the third FSM run simultaneously in parallel and are mutually independent. They both run until one of them fails or one of them succeeds in finding its designated component bit pattern.

DEPR:
In this manner, the speed of detection of a characteristic data pattern is increased, the number of components is decreased, and effective, real time control can be achieved for high speed data networks.

DEPR:
Still another feature of the adaptive, active monitor is the programmability of the FSMs and the programmability of their interconnection. Each FSM consists of an address register and a memory. The address register has two portions, an n-X bit wide first portion and a X-bit wide second portion X. X is one bit for binary data, X is a word of two bits for Manchester encoded data, or X is a word of five bits for FDDI encoded data. The X-bit wide portion is connected to the input data stream which contains the characteristic data pattern of interest. The n-X bit wide portion contains data which is output from the memory. The next address to be applied by the address register to the memory is made up of the X-1 bit wide portion and the next arriving X-bit word from the input data stream.

DEPR:
Each memory has a plurality of data storage locations, each having a first portion with n-X bits, to be output to the address register as part of the next address. Many of the memory locations have a second portion which stores a command to reset the address register if the FSM fails to detect its designated component bit pattern.

DEPR:
A terminal location in the memory of an FSM will include a start signal value to signal another FSM to start analyzing the data stream. If the terminal location in a predecessor FSM memory is successful in matching the last bit of its designated component bit pattern, then it will output a starting signal to a succeeding FSM. The succeeding FSM will begin analyzing the data stream for the next component bit pattern of the characteristic data pattern. The memory of an FSM can be a writable RAM, enabling its reconfiguration to detect different component bit patterns.

DEPR:
Another feature contributing to the programmability of the adaptive, active monitor is the inclusion of a programmable cross point switch, which enables the starting signals from predecessor FSMs to be directed to different successor FSMs. This enables changing the order and combination of FSMs per forming analysis of a bit stream, to detect differently organized characteristic data patterns.

DEPR:
Another feature of the adaptive, active monitor is its functioning in an information collection architecture, to monitor the traffic on a network and to provide event counts for the occurrence of data patterns which are used to control the characteristics of the network.

DEPR:
Further, diverse sized characteristic data patterns can be detected. For example, if when monitoring the 10-bit pattern it is determined that more than 10 bits of information are required, the adaptive monitor feature may be dynamically altered to change the length of the pattern that can be detected. This ability provides increased insight into the characteristics of the data stream.

DEPR:
Another feature of the adaptive, active monitor is its ability to receive serial data streams which include serial patterns of multiple state symbols such as in token ring networks and in fiber optical distributed data interface (FDDI) networks.

DEPR:
An additional feature of the adaptive, active monitor is an information collection architecture system, with an adaptable, simultaneously parallel array of finite state machines, for monitoring a data communications network. The system includes an array of at least three finite state machines, embodied on a VLSI circuit chip or alternately in separate task partitions of a multitasked data processor. Each finite state machine in the array, includes a memory, an address register coupled to the network, a start signal input and a pattern detection output coupled to a counter, the memory thereof storing a finite state machine definition for detecting a unique data pattern on the network. Each machine can detect a different pattern. A programmable interconnection means is coupled to the finite state machines in the array, for selectively interconnecting the pattern detection outputs to the start signal inputs. An event vector assembly means, has inputs coupled to the counters, for assembling an event vector from an accumulated count value in the counters, representing a number of occurrences of the data patterns on the network. An information collection means, has an input coupled to the event vector assembly means, an array output coupled to the memories of the finite state machines, and a configuration output coupled to the programmable inter connection means, for receiving the event vector and in response thereto, changing the array to change data patterns to be detected on the network.

DEPR:
The information collection means, in response to receiving the event vector, changes a first interconnection arrangement of the first pattern detection output being connected to the second start signal input, to a second interconnection arrangement of the first pattern detection output being connected to the third start signal input. This changes the composite pattern to be detected. Further, the information collection means, in response to receiving the event vector, changes a first interconnection arrangement of the first pattern detection output being connected to the second start signal input, to a second interconnection arrangement of the first pattern detection output being connected to both the second start signal input and to the third start signal input. This creates a simultaneous, parallel finite state machine

operation. Further, the information collection means, in response to receiving the event vector, outputs new finite machine definition data to at least the first memory to change the first data pattern to be detected.

DEPR:
Further, the information collection means is coupled to the network, and in response to receiving the event vector, outputs a control signal to the network to alter communication characteristics thereof. The resulting information collection architecture system provides a flexible, rapidly reconfigurable means to monitor and control data communications networks, through real time monitoring of the data patterns in their traffic.

DEPR:
The principle of the adaptive active monitoring invention is shown in FIGS. 1A-1F. The adaptive active monitoring invention, monitors a serial data stream having a characteristic pattern. It detects characteristic data patterns in messages on a high speed data network, starting delimiters, tokens, various types of frames such as a MAC frame, a beacon frame, message frames, etc., and other protocol information. Such data streams typically include serial patterns of binary bits. However, some communications protocols, such as the IEEE 802.5 token ring protocol, have multiple state symbols such as the J, K, 0 and 1 symbols (four states), and they can also be accommodated by the invention. The IEEE 802.5 token ring protocol is described in the IEEE Standard 802.5, token ring access method, available from IEEE Incorporated, New York, N.Y., 1989.

DEPR:
Characteristic data patterns such as these include component bit patterns, some of which are common among several characteristic data patterns. For example, a starting delimiter bit pattern is a common component which begins many other characteristic data patterns such as a token, an ending delimiter abort (EDAB) and other communication messages in a token ring protocol. The occurrence of multiple bit patterns in a characteristic data pattern can be generalized by referring to a first component pattern which is immediately followed by a second component pattern for a first characteristic data pattern. A second characteristic data pattern can employ the same first component pattern which will then immediately be followed by a third component pattern which is different from the second component pattern.

DEPR:
In protocols having two characteristic data patterns with some of the component bit patterns being the same, the objective of pattern detection will be to detect either one of the two characteristic data patterns. In accordance with the adaptive active monitoring invention, the predecessor finite state machine will have a plurality of successor finite state machines which run simultaneously and parallel. The predecessor finite state machine will send a starting signal to both of the successor finite state machines, when the predecessor finite

state machine has successfully detected the first component data pattern. The starting signal initializes both of the successor finite state machines to take over the analysis of the bit stream which follows the first component pattern, in order to look for the second component bit pattern or alternately the third component bit pattern. Both successor finite state machines run simultaneously and parallel and are mutually independent. They both run until one of them fails or one of them succeeds in finding its designated component bit pattern. In this manner, the speed of detection of a characteristic data pattern is increased, the number of components of the finite state machine array is decreased, and the effective real time control can be achieved for high speed data networks.

DEPR:
Turning now to FIG. 1A, a parallel finite state machine adaptive monitor 100 is shown. The bit stream 124 which comes from the communications network, is commonly connected to the input of all of the finite state machines FSM 130(J), 130(O), 130(H), 130(N), 130(I), 130(M), 130(E) and 130(B). The starting signal 202 applied to a finite state machine for example FSM 130(J), comes from the termination signal 208 generated by another finite state machine in the array 100. For example, the finite state machine FSM 130(O) has its start signal 202(O) derived from the starting signal 202(0) output by the finite state machine 130(J). The interconnection of the output termination signal 208 of a predecessor finite state machine, to the starting signal input 202 of a successor finite state machine, is accomplished by the programmable cross point switch 210 shown in FIG. 1A. The cross point switch 210 is configured to interconnect the starting signal input of a successor finite state machine to the termination signal output of the predecessor finite state machine, in order to accomplish a desired sequential analysis of component bit patterns making up consecutive portions of a data pattern of interest.

DEPR:
FIG. 1A also shows the pattern alarms 144, 144' and 144" which result from the satisfactory completion of the analysis of a corresponding characteristic data pattern. The finite state machines FSM 130(J) through FSM 130(B) in FIG. 1A, can be each embodied as a large scale integrated circuit (LSI) chip, connected by means of a bus for conducting the start signals 202 and the termination signals 208 with the programmable cross point switch 210. The programmable cross point switch 210 can also be a separate LSI circuit chip. In another embodiment of the invention, the finite state machines FSM 130(J) through 130(B) and the programmable cross point switch 210, can collectively be integrated into a very large integrated circuit VLSI circuit chip.

DEPR:
FIG. 1B shows another embodiment of the finite state machine array 100 shown in FIG. 1A, wherein it provides for a large array of selectively interconnectable finite state machines. In FIG. 1B, the programmable cross point switch 210 can

selectively interconnect the termination signals from predecessor finite state machines to the start signal inputs 202 of successor finite state machines in a flexible, programmable manner. By configuring the interconnection pattern in the cross point switch 210, predecessor finite state machines may be selected for applying starting signals to successor finite state machines. A particular pattern of interconnection is shown in FIG. 1E and another particular pattern is shown in FIG. 1F. The embodiment shown in FIG. 1B can also be implemented as a plurality of LSI circuit chips or alternately all of the elements shown in FIG. 1B can be integrated onto the same very large scale integrated circuit chip.

DEPR:
Turning now to FIG. 2, it is seen how the finite state machine array is interconnected to perform three parallel data pattern analyses in the bit stream 124. The frame detector 110 consists of the FSM 130(F), 130(R), 130(A), 130(M) and 130(E'). These finite state machines look for the consecutive characters "F", "R", "A", "M" and "E" in the bit stream 124. If they are found, then the start signal 202(J) is passed from the FSM 130(E') to the FSM 130(J).

DEPR:
The frame type detector 112, which consists of the FSM 130(J), 130(O), 130(H), 130(N), 130(I), 130(M), 130(E) and 130(B), performs frame type detection for three frame type character patterns in the bit stream 124, which can immediately follow the "FRAME" frame designation in the bit stream 124. In accordance with the invention, the FSM 130(J) outputs three starting signals, 202(O), 202(I), and 202(E), to start three parallel, simultaneous, independently operating finite state machine sequences shown in FIG. 2. In the example shown in FIG. 2, the frame detector 110 searches for the character pattern "FRAME." If that pattern is found in the bit stream of 124, then the FSM 130(J) starts the three parallel simultaneous sequences which look for the character pattern "JOHN", or "JIM", or "JEB". The output alarm "FRAME.sub.-- JOHN", is output from the FSM 130(N) if that character pattern is identified. The output alarm "FRAME.sub.-- JIM", is output from the FSM 130(M), if that character pattern "JIM", is found in the bit stream 124. The output alarm "FRAME.sub.-- JEB", is output from FSM 130(B), if "JEB" is found in the bit stream 124. Each of these frame type character patterns must follow the first pattern of "FRAME".

DEPR:
FIG. 3 illustrates a security agent (SA) 10 and communicating devices 40 and 41 connected to a bit stream 124. SA 10 monitors bit stream 124 searching for characteristic patterns in data transmitted between communicating devices 40 and 41. In response to a detected characteristic pattern, SA 10 modifies, injects, or deletes information in bit stream 124. For example, SA 10 can produce and transmit a security alert message to one or both of the communicating devices 40 and 41 or to some other device such as a network security manager not shown in FIG. 3.

The SA 10 can also send a response message to one of the communicating devices 40 and 41 or to some other device such as a network security manager via a different communication path not shown in FIG. 3. Those skilled in the art will recognize that many possible paths may exist for the SA 10 to transmit responses to other devices or to take some action in response to a detected characteristic pattern by modifying, injecting, or deleting information in bit stream 124.

DEPR:
FIG. 4 illustrates a security agent (SA) 10 connected to a bit stream 124 consisting of an adaptive, active monitor 100 and a responder 300. Adaptive, active monitor 100 detects characteristic data patterns in bit stream 124 which, in turn, causes a pattern alarm signal 144 to be output. Pattern alarm signal 144 activates responder 300, which in turn responds by modifying, injecting, or deleting information in bit stream 124. Further, in accordance with the invention, adaptive, active monitoring means 100 may be capable of detecting a plurality of characteristic patterns, in which case responder 300 is capable of modifying, injecting, and deleting information in bit stream 124 in a plurality of ways, depending on, and in response to said plurality of pattern alarm signals 144, 144', 144" not shown in FIG. 4.

DEPR:
Referring to FIG. 5, one of a plurality of pattern alarm signals (144, 144', ..., 144") is encoded by pattern alarm encoder 301, whereupon the encoded output is stored in one of a plurality of non-volatile registers 303 and a program latch 302 is set. The program latch is used as a means to signal responder 300 to process the received pattern alarm signal. Non-volatile registers 303 contains a set of parameter values that collectively represent those values used to initialize or configure responder 300, such as (1) a security agent identifier which can be used to uniquely identify messages originated by the security agent, (2) an encoded pattern alarm signal identifying the current alarm signal to be processed by responder 300, a (3) sequence number counter that is incremented and transmitted in each message originated by the security agent, and (4) a secret cryptographic key used by the data encryption algorithm 304 to encrypt/decrypt messages. Data encryption algorithm 304 can be any of several cryptographic algorithms such as the Data Encryption Algorithm (DEA) described in American National Standard X3.92-1981, DATA ENCRYPTION ALGORITHM, American National Standards Institute, New York (Dec. 31, 1981). Security alert message transmission means 306 performs the function of transmitting a constructed security alert message over bit stream 124 using a defined protocol and access method. For example, the constructed security alert message can be inserted into bit stream 124, In a token ring, the security alert message transmission means 306 can consist of software and hardware components permitting the transmission of a "unit" of data on the token ring network. On the other hand, security alert message transmission means 306 can transmit a security alert message by intercepting and modifying data within an existing data structure transmitted

over bit stream 124. Upon detecting that program latch 302 has been set (e.g., by polling program latch 302 during periods when processor 305 is inactive), processor 305 constructs a security alert message from information stored in non-volatile registers 303 and causes a message authentication code to be calculated on the security alert message by invoking data encryption algorithm 304, passing the so-produced security alert message and message authentication code to security alert message transmission means 306. Security alert message transmission means 306 causes the security alert message and message authentication code to be transmitted via bit stream 124 to a destination device such as a network security manager device.

DEPR:
Processor 305 can consist of a microprocessor, microcode maintained in a read only memory (ROM), and a random access memory (RAM) for storage of intermediate results. In like manner, processor 305 can consist of a collection of parallel and serial finite state machines similar to the Hershey adaptive, active monitor 100. However, microprocessor-ROM-RAM implementation can be most advantageous since unlike the adaptive, active monitor 100, responder 300 is likely to be required to process pattern alarm signals only occasionally in response to detected security events. That is, security events are likely to arrive only after relatively long periods of delay. Thus, an implementation based on parallel FSMs can best be characterized as "over-kill" with respect to responder 300.

DEPR:
FIG. 6 is a block diagram illustration of a Responder 300 that processes security events, which are characterized as one of a plurality of possible pattern alarms (144, 144', ..., 144") output by the Adaptive, Active Monitor 100 of FIG. 4. FIG. 6 is a further elaboration of FIG. 5 wherein nonvolatile registers 303 consist of a security agent identifier 321, a security code 322, a sequence number counter 323, and a cryptographic key 325 and processor 305 consists of a security alert message (SAM) production means 331, a message authentication code (MAC) production means 332, and storage for a so-produced security alert message (SAM) 341 and a so-produced message authentication code (MAC) 342.

DEPR:
Referring to FIG. 6, one of a plurality of pattern alarm signals (144, 144', ..., 144") is encoded by pattern alarm encoder 301, whereupon the encoded output is stored in one of a plurality of non-volatile registers 303 and a program latch 302 is set. Upon detecting that program latch 302 has been set (e.g., by polling program latch 302 during periods when processor 305 is inactive), processor 305 performs the following steps. Sequence number counter 325 is incremented and security agent identifier 321, security code 322, and sequence number counter 325 are read from non-volatile registers 303 and processed by security alert message (SAM) production means 331 to produce an output security alert message 341. SAM 341 is next passed to message authentication code (MAC) production

means 332 to produce a message authentication code (MAC) 342. In order for Message authentication code (MAC) production means 332 to produce MAC 342, SAM 341 and cryptographic key 327, which is retrieved from non-volatile registers 303, are passed as inputs to data encryption algorithm 304 which performs the individual steps of encryption in order to produce MAC 342. The so-produced SAM 341 and MAC 342 are the provided to security alert message transmission means 306 which causes SAM 341 and MAC 342 to be transmitted via bit stream 124 to a network security manager device.

DEPR:
FIG. 7 depicts a security alert message 341 consisting of a security agent identifier 321, a security code 322 and a sequence number counter 325. Further in accordance with the invention, security code 322 consists of a CATEGORY 323 which provides a broad characterization of the detected pattern alarm and a TYPE 324 which specifies a particular pattern alarm within category 323. For example, the security alert message can define categories such as these (1) virus, (2) inappropriate word usage, (3) intrusion, and (4) non-encrypted text. For CATEGORY = virus, TYPE can be defined as (1) Christmas EXEC, (2) ..., and so forth. Those skilled in the art will recognize (1) there are many ways in which security events can be divided into categories and types and (2) said invention is independent of the categories and types which are selected.

DEPR:
A security agent identifier 321 is included in security alert message 341 so that the messages' receiver will have proof of the identity of the security agent who has detected a pattern alarm and originated the security alert message. A sequence number counter 325 is included in security alert message 341 so that an adversary, who may intercept and replay security alert message 341 in bit stream 124, will be unable to cause the designated receiver to accept the security alert message as genuine. In an alternate embodiment of the invention, a time stamp can be used in place of sequence number counter 325 to prevent message replay attacks.

DEPR:
FIG. 8 is a block diagram illustration of an alternate embodiment of the invention (as described in FIG. 6) wherein the pattern alarms 144a, 144b, ..., 144n from the Hershey adaptive, active monitor 100 are applied to counters 360a, 360b, ..., 360n, respectively, in order to prevent adaptive, active monitor 100 of FIG. 4 from over-running responder 300. The sizes of counters 360a, 360b, ..., 360n are set so that the number of pattern alarm signals does not cause a counter overflow during the interval of time in which a security alert message is produced and transmitted. For example, 32-bit counters are more than adequate to prevent counters from overflowing.

DEPR:
Referring now to FIG. 8, each pattern alarm signal 144i causes its associated counter 144i to be incremented by value +1, so

that each counter records the numbers of respective pattern alarm signals produced by adaptive, active monitor 100. Processor 305 contains three processing functions, as follows: (1) counter scanning means 333, (2) security alert message production means 331, and (3) message authentication code production means 332. Counter scanning means 333 continually scans the counters, 360a, 360b, etc., searching for a non-zero counter value. When the final counter 360n is reached, the scanning continues with counter 360a, and so forth. When a non-zero counter value is detected, the counter value is read out and a security alert message and message authentication code are produced and transmitted. It is assumed that the process of reading out a counter value causes the counter to be reset to zero. In this way, the counter can continue to be updated during the time interval when a security alert message and message authentication code are produced and transmitted. Afterwards, counter scanning means 333 continues searching for a non-zero counter value--starting with the next counter in sequence following the counter that was just processed. Counter scanning means 333 also makes use of an index value representing the index of the counter currently being scanned. Upon detecting a non-zero counter value (via counter scanning means 333), processor 305 performs the following steps:

DEPR:
6. Security alert message production means 331 produces a security alert message 341, conforming to the message format shown in FIG. 9, consisting of a security agent identifier 321, a security code 322, a sequence number counter 325, and a pattern alarm counter value 326 corresponding to the value in counter 360i.

DEPR:
In yet another alternative embodiment of the invention (not shown in a figure), responder 300 can periodically read out all the counter values and form a single security alert message containing the counter values corresponding to each pattern alarm. In this case, responder 300 would require an internal clock, e.g., a counter that is incremented by 1 for each bit in bit stream 124 that is scanned by adaptive, active monitor 100 of FIG. 4. When the clock reaches a predetermined threshold value, processor 305 would gain control. Processor 305 would then read the counter values, produces a security alert message containing a vector of counter values and a message authentication code, and transmits the security alert message and message authentication code to the network security manager via bit stream 124.

DEPR:
FIG. 9 depicts an extended security alert message 341 consisting of a security agent identifier 321, a security code 322, a sequence number counter 325, and a pattern alarm counter value 326. The extended security alert message 341 of FIG. 9 differs from the security alert message 341 of FIG. 7 in that the extended security alert message 341 of FIG. 9 contains a pattern alarm counter value 326. Pattern alarm counter value 326 represents the number of pattern alarms (with security code

322) detected by security agent 10 of FIG. 4 (with security agent identifier 321). Extended security alert message 341 can also contain a time-stamp instead of a sequence number counter. This would have the added advantage that a network security manager who receives the security alert message can easily calculate a rate (number of occurrences per standard interval of time) at which the security events are occurring.

DEPR:
Those skilled in the art will recognize that the embodiment of FIG. 8 can be easily adapted to detect and respond to security events corresponding to so-called "dirty" words (4-letter words). In that case, a first "dirty" word is associated with pattern alarm 144a, a second "dirty" word is associated with pattern alarm 144b, and so forth. The security code 322 would consist of a category = "dirty words" and a type indicating the particular "dirty" word to be reported. The pattern alarm counter value 326 would specify the number of such words detected.

DEPR:
FIG. 10 is a block diagram illustration of another alternate embodiment of the invention wherein the Hershey adaptive, active monitor 100 of FIG. 4 is configured as an intrusion detector and responder 300 is designed to produce and transmit a security alert message whenever the number of detected pattern alarms of a particular type in a given interval of time reaches a prescribed threshold value. Referring to FIG. 10, the Hershey adaptive, active monitor 100 of FIG. 4 is configured to scan bit stream 124 for three characteristic patterns, specified in double quotation marks:

DEPR:
Each of these characteristic patterns represents a particular system response in an incorrect and invalid login. The phrase "PASSWORD NOT AUTHORIZED" (in Extended Binary Coded Decimal Interchange Code) is an IBM VM message provided to a host-attached workstation if the login sequence fails. The user VM login screen has the following prompts: "USERID ===V" and "PASSWORD ===>," specified in double quotation marks. These standard phrases can also be defined as characteristic patterns, except that in this case the Hershey adaptive, active monitor 100 would track both invalid as well as valid login requests. The phrase "You entered an invalid login name or password." is a standard response in UNIX to a failed login. In this case, the UNIX login screen has the following prompts: "login:" and "Password:". These standard phrases can also be defined as characteristic patterns. In IBM's version of TCP/IP File Transfer Protocol (FTP) for VM, the system prompts the user with the following: "USER (identify yourself to the host):". In the event of a failed login, the phrase "Login incorrect." is displayed to the user. Thus, the characteristic patterns specified in FIG. 10 will detect failed login attempts for (1) VM, (2) UNIX, and TCP/IP FTP for VM. The reader will appreciate that similar characteristic patterns can be specified for other operating systems and systems applications requiring user login. Referring again to FIG. 10, the

characteristic data "PASSWORD NOT AUTHORIZED" is associated with pattern alarm 144a, the characteristic data "You entered an invalid login name or password." is associated with pattern alarm 144b, and the characteristic data "Login incorrect." is associated with pattern alarm 144c. In like manner, pattern alarms 144a, 144b, and 144c are uniquely associated with counters 360a, 360b, and 360c, respectively. For example, counter 360a contains a value representing the number of pattern alarms 144a (corresponding to "PASSWORD NOT AUTHORIZED") received since said counter was last reset to zero. The counters themselves are assumed to be large enough so that they will not overrun. 32-bit counters would be sufficient to prevent such an overrun. In like manner, counter 360b contains a value representing the number of pattern alarms 144b (corresponding to "You entered an invalid login name or password.") received since said counter was last reset to zero and counter 360c contains a value representing the number of pattern alarms 144c (corresponding to "Login incorrect.") received since said counter was last reset to zero.

DEPR:
Responder 300 also contains a clock 350 which is attached to Hershey adaptive, active monitor 100 via line 372. Clock 350 is an incrementing counter. For each bit sampled in bit stream 124, a signal is sent via line 372 to clock 350, which causes the clock to increment by +1. When clock 350 reaches a predefined threshold value (e.g., the counter has a high-order one bit), program latch 302 is set. A clock size and threshold value are selected so that the time it takes to produce and transmit a security alert message is less than the time it takes clock 350 to cycle from zero to its threshold value. When processor 305 is not busy producing and transmitting a security alert message, processor 305 is busy monitoring program latch 302. When processor 305 detects that program latch 302 has been set, it reads the counter values (360a, 360b, and 360c), resets the counters to zero, and resets program latch 302. The counters are read and reset before the Hershey adaptive, active monitor is able to send another pattern alarm 144, thus preventing loss of information. Once the counters (360a, 360b, and 360c) have been read and reset, Hershey adaptive, active monitor continues, as before, sending pattern alarms (144a, 144b, and 144c). Processor 305 performs the following steps:

DEPR:
The Hershey, adaptive active monitor 100 is configured to scan for each of the 256 possible 8-bit characters in the data portion of each transmitted frame. The Hershey adaptive active monitor 100 accomplishes this scanning for the starting and ending delimiters for each the data block and then scanning and recording each character within each data block. A method for accomplishing this is taught by Hershey and Waclawsky in copending U.S. patent application entitled "System and Method for Adaptive, Active Monitoring of a Serial Data Stream Having a Characteristic Pattern," Ser. No. 08/138,045 cited above under Related Patents and Patent Applications. Responder 300 is designed to produce and transmit a security alert message whenever the distribution of detected characters in a given

interval of time "looks" more like plaintext than ciphertext, which is based on a statistical calculation.

DEPR:
Referring to FIG. 11, the Hershey, adaptive, active monitor 100 of FIG. 4 is configured to scan bit stream 124 for any of the 256 characters within the data portion of a transmitted frame. Pattern alarm 144a corresponds to the 1st character, designated B'00000000'; pattern alarm 144b corresponds to the 2nd character, designated B'00000001', ..., pattern alarm 144n corresponds to the 256th character, designated B'11111111'. For a given interval of time, counter 360a records the number of detected characters of the form B'00000000', counter 360b records the number of detected characters of the form B'00000001', ..., counter 360n records the number of detected characters of the form B'11111111'.

DEPR:
Responder 300 also contains an accumulator 361, which is connected to each of the 256 pattern alarm lines (144a, 144b, 144n). Each pattern alarm causes its associated counter to increment by +1 and it also causes the accumulator 361 to increment by +1. In this manner, accumulator 361 contains a value equal to the sum of the values in the 256 counters (360a, 360b, ..., 360n). When accumulator 361 reaches a predefined threshold value, program latch 302 is set. An accumulator size and threshold value are selected so that enough characters are sampled from the data portion of the frame or frames to allow a meaningful statistic to be calculated. For example, a few hundred characters would ordinarily suffice to discriminate plaintext from ciphertext, although a value of 1000 shall be selected in order to illustrate the process to be used. When processor 305 is not busy producing and transmitting a security alert message, processor 305 is busy monitoring program latch 302. When processor 305 detects that program latch 302 has been set, it reads the counter values (360a, 360b, ..., 360n), resets the counters to zero, and resets program latch 302. The counters are read and reset before Hershey adaptive, active monitor is able to send another pattern alarm 144, thus preventing loss of information. Once the counters (360a, 360b, ..., 360n) have been read and reset, Hershey adaptive, active monitor continues, as before, sending pattern alarms (144a, 144b, ..., 144n).

DEPR:
In the situation where ONLY compressed data is transmitted in bit stream 124, it may be possible to adapt the above test to distinguish compressed plaintext from ciphertext. However, this may depend on the quality of the compression being used and it may also depend on a knowledge of the compression algorithm itself in order to construct a testing algorithm that is assured of success. For example, it may be necessary for adaptive, active monitor 100 to be configured to scan for diagrams (2-letter groups) or trigrams (3-letter groups), since the compression algorithm may be very good at "flattening" the frequency distribution of single letters. That is, a test involving only single letters may be insufficient to

distinguish compressed plaintext from ciphertext, at least for sample sizes which are considered practical. For example, it would do little good if one could distinguish compressed plaintext from ciphertext if it required a sample size of 10 million characters. But by knowing something about the compression algorithm itself, one has a much better chance of constructing a sampling and testing algorithm to detect differences between compressed plaintext and ciphertext. Those skilled in the art will also recognize that from one perspective the goal of compression techniques is to remove the redundancy commonly associated with any ordinary natural language such as English. The more redundancy that one can remove from a compressed text the more it looks like random text or ciphertext. However, an excellent encryption algorithm such as the Data Encryption Algorithm will produce ciphertext that is hardly distinguishable from random text. On the other, a good compression algorithm will produce compressed text that will generally have some, even a small amount, of redundancy left in it. Hence, the compressed text will not look as much like random text as the ciphertext produced with the DEA will look like random text. This small difference between compressed text and ciphertext is enough to be distinguished provided a large enough sample size is used. In theory, one can distinguish compressed text from ciphertext if a large enough sample size is used. In practice, one can distinguish compressed text from ciphertext only if the compression algorithm is a poor one or some feature of the compression algorithm can be exploited to allow the testing algorithm to operate with only small or modest sample sizes. In summary, the testing procedure described above can in some situations (depending on the compression algorithm) be used to distinguish compressed plaintext from ciphertext--provided a large enough sample size is used. In other cases, the adaptive action monitor 100 must be reconfigured to scan for particular patterns (such a diagrams and trigrams).

DEPR:
FIG. 12 is an example embodiment of pattern alarms and counters of FIG. 8 configured for virus detection. Referring to FIG. 12, the pattern alarms 144a, 144b, ..., 144n originating with adaptive, active monitor 100 of FIG. 4 are each uniquely associated with a particular characteristic viral pattern. Adaptive, active monitor 100 scans bit stream 124 for these virus patterns, which are character strings or patterns that uniquely identify each virus. As in FIG. 8, the pattern alarms 144a, 144b, ..., 144n from the Hershey adaptive, active monitor 100 are applied to counters 360a, 360b, ..., 360n, respectively. In this way, a clever adversary can not flood the network with viral agents hoping that some will escape detection by over running responder 300. The sizes of counters 360a, 360b, 360n are set so that the number of pattern alarm signals does not cause a counter overflow during the interval of time in which a security alert message is produced and transmitted. For example, 32-bit counters are more than adequate to prevent counters from overflowing. Otherwise, the processing steps to handle a detected virus are the same as those already described in and for FIG. 8.

DEPR:
Those skilled in the art will recognize that the viral patterns described in FIG. 12 may be static patterns or dynamically configurable patterns, depending on how adaptive, active monitor 100 is designed. The Hershey adaptive, active monitor is capable of dynamic re-configuration, thus enabling the security agent 10 to be updated in real time to detect the presence of a suspected offending virus. Following such a re-configuration, which will also cause the type field associated with security code 322 to be adjusted so that each pattern alarm is uniquely associated with a corresponding virus type, both the adaptive, active monitor 100 and responder 300 of security agent 10 of FIG. 4 will operate normally as before.

DEPR:
An example embodiment of pattern alarms and counters for FIG. 8 configured for virus detection, is shown in FIG. 12.

DEPV:
5. The value of index i is then used as a means to access a corresponding security code. It is assumed that processor 305 contains a table of n predefined security codes corresponding to the n pattern alarm signals 144a, 144b, ..., 144n and to the n counters 360a, 360b, ..., 360n, respectively. Thus, each index value uniquely specifies a security code 322 consisting of a category 323 and a type 324.

CLPR:
2. The system of claim 1, wherein said responding means, in response to receiving said security event vector, changes a first interconnection arrangement of said first security threat pattern detection output being connected to said second start signal input, to a second interconnection arrangement of said first security threat pattern detection output being connected to said third start signal input.

CLPR:
3. The system of claim 1, wherein said responding means, in response to receiving said security event vector, changes a first interconnection arrangement of said first security threat pattern detection output being connected to said second start signal input, to a second interconnection arrangement of said first security threat pattern detection output being connected to both said second start signal input and to said third start signal input, for simultaneous, parallel finite state machine operation.

CLPR:
4. The system of claim 1, wherein said responding means, in response to receiving said security event vector, outputs new finite machine definition data to at least said first memory to change said first data security threat pattern to be detected.

CLPR:
17. The system of claim 1, wherein said security threat is the

occurrence of a natural language pattern.

CLPR:
18. The system of claim 1, wherein said security threat is an intrusion detection pattern.

CLPR:
21. The system of claim 20, wherein in response to receiving said security event vector, changing a first interconnection arrangement of said first security threat pattern detection output being connected to said second start signal input, to a second interconnection arrangement of said first security threat pattern detection output being connected to said third start signal input.

CLPR:
22. The method of claim 20, wherein in response to receiving said security event vector, changing a first interconnection arrangement of said first security threat pattern detection output being connected to said second start signal input, to a second interconnection arrangement of said first security threat pattern detection output being connected to both said second start signal input and to said third start signal input, for simultaneous, parallel finite state machine operation.

CLPR:
23. The method of claim 20, wherein in response to receiving said security event vector, outputting new finite machine definition data to at least said first memory to change said first data security threat pattern to be detected.

CLPR:
36. A method for information collection by adaptive, active security monitoring of a serial stream of data having a characteristic virus pattern including a first occurring and a second occurring virus pattern portions, comprising the steps of:

CLPR:
40. A method for adaptive, active security monitoring of a serial stream of data having a characteristic virus pattern including a first occurring and a second occurring virus pattern portions, comprising the steps of:

CLPR:
42. An information collection architecture system for adaptive, active security monitoring of a serial stream of data having a characteristic virus pattern including a first occurring and a second occurring virus pattern portions, for performing security monitoring and control operations on a data communications medium providing said data stream, comprising:

CLPR:
45. The system of claim 42, wherein said characteristic virus pattern is from a fiber optical distributed data interface (FDDI) data communications medium.

CLPV:
a first finite state machine in said array, including a first memory, a first address register coupled to said network, a first start signal input and a first security threat pattern detection output coupled to a first counter, said memory thereof storing a first finite state machine definition for detecting a first data security threat pattern on said network;

CLPV:
a second finite state machine in said array, including a second memory, a second address register coupled to said network, a second start signal input and a second security threat pattern detection output coupled to a second counter, said memory thereof storing a second finite state machine definition for detecting a second data security threat pattern on said network;

CLPV:
a third finite state machine in said array, including a third memory, a third address register coupled to said network, a third start signal input and a third security threat pattern detection output coupled to a third counter, said memory thereof storing a third finite state machine definition for detecting a third data security threat pattern on said network;

CLPV:
a programmable interconnection means coupled to said first, second and third finite state machines, for selectively interconnecting said first security threat pattern detection output to at least one of said second and third start signal inputs;

CLPV:
a security event vector assembly means, having inputs coupled to said first, second and third counters, for assembling a security event vector from an accumulated count value in said first counter and at least one of said second and third counters, representing a number of occurrences of said first data security threat pattern and at least one of said second and third data security threat patterns on said network; and

CLPV:
a responding means, having an input coupled to said security event vector assembly means, an array output coupled to said memory of said first, second and third finite state machines, and a configuration output coupled to said programmable interconnection means, for receiving said security event vector and in response thereto, changing said array to change data security threat patterns to be detected on said network.

CLPV:
storing a first finite state machine definition for detecting a first data security threat pattern on said network, in a first finite state machine in said array, including a first memory, a first address register coupled to said network, a first start

signal input and a first security threat pattern detection output coupled to a first counter;

CLPV:
storing a second finite state machine definition for detecting a second data security threat pattern on said network, in a second finite state machine in said array, including a second memory, a second address register coupled to said network, a second start signal input and a second security threat pattern detection output coupled to a second counter;

CLPV:
storing a third finite state machine definition for detecting a third data security threat pattern on said network, a third finite state machine in said array, including a third memory, a third address register coupled to said network, a third start signal input and a third security threat pattern detection output coupled to a third counter;

CLPV:
selectively interconnecting said first security threat pattern detection output to at least one of said second and third start signal inputs;

CLPV:
assembling a security event vector from an accumulated count value in said first counter and at least one of said second and third counters, representing a number of occurrences of said first data security threat pattern and at least one of said second and third data security threat patterns on aid network; and

CLPV:
receiving said security event vector and in response thereto, changing said array to change data security threat patterns to be detected on said network.

CLPV:
accessing a first-addressable memory having a plurality of data storage locations, each having a first portion with n-x bits, said first memory having an n-bit address input coupled an output of said first address register, said first memory configured with data stored in first and second ones of said data storage locations to represent a first digital filter for said first occurring virus pattern;

CLPV:
outputting a start signal from said second one of said data storage locations of said first memory having a start signal value stored therein, which is output when said first occurring portion of said characteristic virus pattern is detected by said digital filter;

CLPV:
accessing a second addressable memory having a plurality of data storage locations, each having a first portion with p-x bits, said second memory having a p-bit address input coupled

an output of said second address register, said second memory configured with data stored in first and second ones of said data storage locations to represent a second digital filter for said second occurring virus pattern;

CLPV:
outputting a security alarm value from said second one of said data storage locations of said second memory having a virus pattern security alarm value stored therein, which is output when said second portion of said characteristic virus pattern is detected by said second digital filter;

CLPV:
counting occurrences of said second portion of said characteristic virus pattern in said data stream, with a counter coupled to said virus pattern security alarm value output, and outputting a count value as an event vector; and

CLPV:
reconfiguring said first addressable memory by storing new data in said first and second ones of said data storage locations to represent a third digital filter for a third occurring virus pattern, in response to said event vector.

CLPV:
accessing a first addressable memory having a plurality of data storage locations, each having a first portion with n-x bits, said first memory having an n-bit address input coupled an output of said first address register, said first memory configured with data stored in first and second ones of said data storage locations to represent a first digital filter for said first occurring virus pattern;

CLPV:
outputting a start signal from said second one of said data storage locations of said first memory having a start signal value stored therein, which is output when said first occurring portion of said characteristic virus pattern is detected by said digital filter;

CLPV:
accessing a second addressable memory having a plurality of data storage locations, each having a first portion with p-x bits, said second memory having a p-bit address input coupled an output of said second address register, said second memory configured with data stored in first and second ones of said data storage locations to represent a second digital filter for said second occurring virus pattern;

CLPV:
outputting an security alarm value from said second one of said data storage locations of said second memory having a virus pattern security alarm value stored therein, which is output when said second portion of said characteristic virus pattern is detected by said second digital filter.

CLPV:

first addressable memory having a plurality of data storage locations, each having a first portion with n-x bits, said first memory having an n-bit address input coupled an output of said first address register, said first memory configured with data stored in first and second ones of said data storage locations to represent a first digital filter for said first occurring virus pattern;

CLPV:
said second one of said data storage locations of said first memory having a start signal value stored therein, which is output when said first occurring portion of said characteristic virus pattern is detected by said digital filter;

CLPV:
second addressable memory having a plurality of data storage locations, each having a first portion with p-x bits, said second memory having a p-bit address input coupled an output of said second address register, said second memory configured with data stored in first and second ones of said data storage locations to represent a second digital filter for said second occurring virus pattern;

CLPV:
said second one of said data storage locations of said second memory having a virus pattern security alarm value stored therein, which is output when said second portion of said characteristic virus pattern is detected by said second digital filter;

CLPV:
a counter coupled to said virus pattern security alarm value output, for counting occurrences of said second portion of said characteristic virus pattern in said data stream, and outputting a count value as an event counter; and